V.Alexiades and A.D. Solomon
Mathematical Modeling of Melting and Freezing Processes
Hemisphere, Washington DC, 1993; ISBN 1-56032-125-3

C H A P T E R 4

NUMERICAL METHODS
THE ENTHALPY FORMULATION

As we have repeatedly remarked, explicit and approximate solutions are obtainable only for simple problems and only in one space dimension. As most realistic phase-change processes do not neatly fall in this category, the mathematical problems modeling such processes may only be attacked numerically.

A mathematical model of a physical process may be thought of as a simulation of the process, i.e. an imitation using mathematical tools. In the same spirit as a laboratory-scale experiment of an industrial process is an imitation of the process by the means and capabilities of the laboratory, a numerical (computer) simulation is an imitation of the process by the means and capabilities of the computer.

Digital computers are capable of representing only a finite number of rational (finite decimal) numbers and therefore can only deal with discrete approximations of continuum concepts such as time and length. Moreover, memory sizes are also finite and small, thus restricting the amount of data that can be processed. Such limited capabilities of computers impose certain limitations and restrictions on the numerical simulation of a physical process. Thus, the physical region must be approximated by a small number of "control volumes," time may vary only in discrete steps, and idealized mathematical concepts, such as derivatives, integrals and limits must be re-approximated by finite-differences, sums and approximate values.

In **§4.1** we explain how such discrete approximations are set up (via finite-differences) for the simplest case of heat conduction without phase change. After a brief discussion of front-tracking methods in **§4.2**, we then quickly turn to the most general and versatile method available for the numerical simulation of phase-change processes, the so-called **enthalpy method**. Its numerical implementation is presented in **§4.3**. The mathematical ideas underlying weak formulations of PDE problems, and the mathematical formulation on which the enthalpy method is based are presented in **§4.4**. Finally, in **§4.5** we establish existence of the weak solution and convergence of the enthalpy scheme to the weak solution.

4.1. NUMERICAL HEAT TRANSFER

4.1.A Introduction

Simulation of a system means imitation of the system by a convenient replacement or ‘‘stand-in,’’ whose performance can be studied in detail. The motivation is to use an inexpensive ‘‘stand-in’’ to tell us what we want to know about the original system. The simulation might consist of a field trial in place of the actual unmonitored process, a laboratory experiment in place of a field trial, or a pencil-and-paper mathematical model in place of the laboratory experiment.

A numerical, or computer, simulation is one in which the ‘‘stand-in’’ for the system is a computer code, whose runs simulate the system’s performance. If the processes taking place are time-dependent, then the computer code must accordingly tell us what is going on with the progress of time. Such a code is often referred to as a ‘‘marching’’ code, with the implication being ‘‘with increasing time.’’

The computer simulation of a time-dependent process rests upon a **discretized** version of a mathematical model of the actual physical process. Thus, continuous quantities, such as energy and temperature, are replaced by their values at discrete points. Time itself is discretized, and the marching process takes place through discrete time steps. Just as the individual frames of a movie must be taken at close enough times, the time steps for a computer simulation must be small enough for us not to lose the impression of continuity.

The truly dramatic advances in digital computer technology achieved over the last 30 years have already elevated Numerical Simulation to the status of a third scientific method, complementing the two traditional methods of Theory and Experiment. Increasingly complicated processes may be realistically simulated numerically, often more effectively and at lower cost than actual experiments, enabling us to better predict, understand and control them. Thus, numerical simulation is fast becoming an indispensable tool in technological discovery and development and a strong driving force in the quantification and mathematization of science and technology. An excellent overview of Numerical Heat Transfer may be found in the Handbook [MINKOWYCZ et al].

There are four basic steps involved in the development of a computer simulation of a physical process:

1. **Determination of the physical problem**. Decide which physical phenomena are important enough to be taken into account, which physical variables define the system, what are the inputs (data), and what is to be found.
2. **Formulation and analysis of the mathematical model**. ‘‘Translate’’ the *physical* problem into a precise *mathematical* problem, identify the data and the unknowns, and convince ourselves that the resulting mathematical problem is well-posed or, at least, that it ‘‘makes sense’’.

3. **Discretization of the problem**. Approximate the problem by a discrete one, i.e. replace all ''continuous'' entities by corresponding ''discrete'' ones, and construct a numerical algorithm for its solution.

4. **Development and implementation of algorithms in a computer code**. Develop algorithms and code embodying them. Check them out and validate the resulting programs.

Consider, for example, heat transfer in a body occupying a region Ω in space. In Step 1, we must decide if heat is transferred by conduction, convection, or radiation; if a phase-change is involved; if temperature alone suffices to describe the thermal state; if the process is transient or steady-state; what are the initial and boundary conditions, etc. In a ''real-life'' situation, many of these decisions may not be as simple as they sound, and various simplifying physical assumptions may be required in order to formulate a ''reasonable'' problem (c.f. **§1.2**). Step 2 is achieved when we determine the equations expressing the physical laws and conditions identified in Step 1. Several examples of this process were presented in CHAPTER **2**. When the resulting mathematical problem is not amenable to analytical treatment, Step 3 becomes necessary, at which time the problem is approximated by a discrete one and algorithms are devised to compute its solution. At Step 4, we write computer programs implementing the algorithms in some convenient computer language, e.g., FORTRAN, and apply them to some simple problems with known solutions (benchmark problems), in order to check that the simulation performs as expected. Clearly, this is an inter-disciplinary endeavor, requiring knowledge from several fields: the scientific discipline pertaining to the process under study, mathematics, numerical analysis, and computer science.

In this section we are concerned with Step 3, for the case of a simple heat-transfer process. Thus, we assume that a well-posed mathematical model of heat-transfer in a region has been formulated, and we discuss its discretization and the construction of effective numerical algorithms for its solution.

Discretization begins with the subdivision of the (spatial) region into ''small'' subregions (**control volumes**), by an imposed spatial grid. The term ''small'' is relative: heat transfer in the ground around a pipe may involve a region tens of feet long; then ''small'' may be inches or feet. On the other hand, for heat transfer in the pipe itself, ''small'' may be a tenth or a hundredth of an inch. Two factors help to determine the size of the control volume. On the one hand, it should be *small* enough to capture essential variations in the computed quantities and to permit us to represent average or typical values as point values. Thus, large temperature gradients require small control volumes, and conversely, small gradients can be captured even by relatively large control volumes. On the other hand, the expense of the resulting numerical computation is the primary limiting factor in how fine a mesh one may use. If unlimited time on a Cray Supercomputer is available to run the code, then the mesh may be a hundred or a thousand times finer than if the code is to be run on a personal computer !

Control volumes are thought of as regions in which ''**local equilibrium**'' is achieved at a time scale considerably shorter than the computational time step;

hence, the value of a field quantity at a nodal point at the center of a control volume may be thought of as representing the **average** of the quantity over the volume.

Having chosen an ''appropriate'' spatial grid, we simulate heat transfer by updating the state of the discrete system through discrete time increments $\Delta t > 0$, using discrete versions of the conservation laws.

There are several actors and inter-related objectives in this play. We want the numerical scheme to be

i) **consistent**, meaning that the discrete *equations* used in the scheme tend to the correct conservation laws as the spatial and temporal grid sizes tend to zero;

ii) **convergent**, meaning that the approximations that it provides to the *solutions* of the (continuum) conservation laws, actually tend to these solutions as the spatial and temporal grid sizes tend to zero;

iii) **stable**, meaning that the computed values at each time-step are relatively insensitive to unavoidable input and roundoff errors;

iv) **effective**, meaning that the scheme achieves the above objectives with minimal computational expense, so that its use in the desired context is affordable.

Certainly, whether or not these objectives can be achieved depends on the discretization method as well as on the method used to solve the discrete equations (and even on the coding itself). The Art and Science of Numerical Computation provides us with several tools and guidelines, and the great advances in computational power and methodology during the last few years already allow us to realistically simulate fairly complicated processes. A useful principle to keep in mind is that simulation is imitation and as such it should try to follow the physical laws as closely as possible.

There are several approaches to the discretization of conservation laws: *finite differences*, *finite elements*, *collocation*, and *spectral* methods. Excellent surveys are given in [ALLEN-HERRERA-PINDER] [MINKOWYCZ et al]; see also [LAPIDUS-PINDER], [DUCHATEAU-ZACHMANN], [SEWELL]. The method that is by far the simplest, easiest to understand and implement, most amenable to direct physical interpretation, and still most widely used is that of finite-differences, especially when derived via **control-volume** discretizations. This is the method that we shall use in this book.

In order to introduce and explain the basic methodology, we begin with the simplest process of heat conduction in a finite slab. As a model problem we treat the following

PHYSICAL PROBLEM: Consider a finite slab, $0 \le x \le l$, with known initial temperature distribution, $T_{init}(x)$. Starting at time $t = 0$, the slab is heated convectively at $x = 0$ (with ambient temperature $T_\infty(t)$ and heat transfer coefficient h), while the back face $x = l$ is kept insulated. We exclude the presence of any volumetric heat sources (see **§4.1.G**). We want to predict the

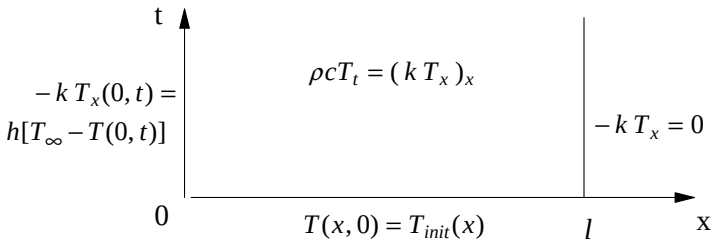

Figure 4.1.1. Model heat transfer problem.

evolution of the temperature field over time. The mathematical formulation is the following.

MATHEMATICAL PROBLEM: Find $T(x,t)$ such that (**Figure 4.1.1**)

$$\rho c T_t = (kT_x)_x, \qquad 0 < x < l, \qquad t > 0 \tag{1a}$$

$$T(x,0) = T_{init}(x), \qquad 0 \le x \le l \tag{1b}$$

$$-kT_x(0,t) = h\,[T_\infty(t) - T(0,t)], \qquad -kT_x(l,t) = 0, \qquad t > 0 \tag{1c}$$

The specific heat, c, thermal conductivity k and heat-transfer coefficient, h, may be known, temperature dependent functions.

4.1.B Control volume discretization of the conservation law

We partition the region of interest into M subregions, called **control volumes**, $V_1, V_2, \ldots, V_M$. With each subregion V_j we associate a **node** x_j, a point inside V_j. We let $\Delta V_j =$ volume of V_j, and $A_{ij} = A_{ji} =$ surface area of the face common to V_i and V_j. For the slab of length l and (constant) cross-sectional area A, we have simply

$$\Delta V_j = A \cdot \Delta x_j \qquad \text{and} \qquad A_{ij} \equiv A, \qquad i, j = 1, \ldots, M, \tag{2a}$$

where Δx_j = length of the jth subinterval, containing node x_j. If we choose to locate nodes at the midpoints of intervals, then the endpoints of the jth subinterval are (**Figure 4.1.2**)

$$x_{j-\frac{1}{2}} = x_j - \frac{\Delta x_j}{2} \quad \text{and} \quad x_{j+\frac{1}{2}} = x_j + \frac{\Delta x_j}{2}, \quad j = 1, \ldots, M, \text{ with } x_{\frac{1}{2}} = 0, \; x_{M+\frac{1}{2}} = l. \tag{2b}$$

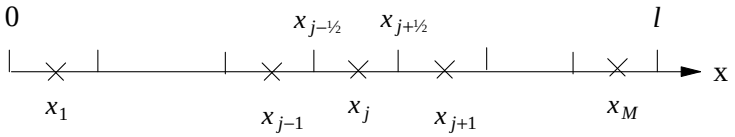

Figure 4.1.2. Nodes and faces of the spatial mesh.

In particular, if the partition is uniform, then $\Delta x_j = \Delta x = l/M$, the nodes x_j are equidistant and

$$x_{1/2} = 0, \quad x_{j-1/2} = (j-1)\Delta x, \quad j = 1, \ldots, M, \quad x_{M+1/2} = M\Delta x = l\ . \tag{2c}$$

For various other common 1- and 2-dimensional meshes see PROBLEMS 3-6.

Let $\Delta t_n > 0$ be time increments and define the discrete time-steps

$$t_0 = 0\ , \quad t_1 = \Delta t_0\ , \cdots , \quad t_{n+1} = t_n + \Delta t_n\ , \cdots , \quad n = 0, 1, 2, \ldots . \tag{2d}$$

If $\Delta t_n = \Delta t$ for all n, then: $t_n = n\Delta t$, $n = 0, 1, 2, \ldots$.

With $T(x,t)$ denoting the exact solution of (1), $T(x_j, t_n)$ represents its value at node x_j at time t_n, and its numerical approximation will be denoted by

$$T_j^n \ \approx\ T(x_j, t_n), \qquad j = 1, \ldots, M, \qquad n = 0, 1, \ldots\ . \tag{3a}$$

We regard T_j^n as also an *approximation* to the *mean temperature of V_j at time t_n*, see PROBLEM 8. In addition, we introduce approximations to the *boundary* temperatures

$$T_0^n \ \approx\ T(0, t_n) \qquad \text{and} \qquad T_{M+1}^n \ \approx\ T(l, t_n), \qquad n = 0, 1, \ldots\ . \tag{3b}$$

From the initial condition (1b),

$$T_j^0 := \ T_{init}(x_j), \qquad j = 1, \ldots, M, \tag{4}$$

is known; for $n = 0, 1, \cdots$, we want to define an algorithm for determining the values T_j^{n+1} at the next time-step, when we know the values T_j^n at the current time-step.

Discrete heat balance

Finite-difference discretizations of the heat equation (1) may be derived in various ways (see [LAPIDUS-PINDER], [PATANKAR], [MINKOWYCZ et al]). We prefer the one that has direct physical meaning, the *discrete heat balance*, that originally formed the basis for the conservation law (1) itself (**§1.2**). Thus, we think of (1) in its primitive form :

$$E_t + q_x \ = \ 0\ , \tag{5}$$

with

$$E \ = \text{thermal energy density per unit volume} = \int_{T_{ref}}^{T} \rho c(\bar{T})\, d\bar{T} \ \approx\ \rho c[T - T_{ref}], \tag{6}$$

T_{ref} being some convenient reference temperature, and

$$q \ = \ \text{heat flux} \ = \ -kT_x \qquad (\text{ Fourier's law }). \tag{7}$$

Note that we may use either the volumetric enthalpy E (per unit volume), or the specific enthalpy e (per unit mass), $E = \rho e$. Integrating (5) over the control volume V_j (**Figure 4.1.3**), and over the time interval $[t_n\ , t_n + \Delta t_n]$, we find

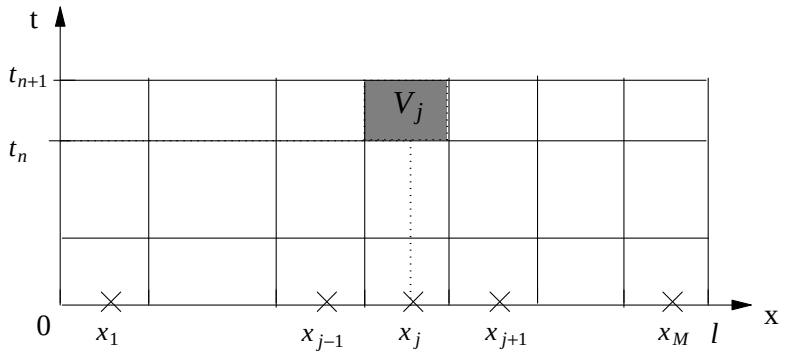

Figure 4.1.3. Space - time grid.

$$\int_{t_n}^{t_{n+1}} \frac{\partial}{\partial t}\left(A \int_{x_{j-1/2}}^{x_{j+1/2}} E(x\,,\,t)\,dx \right) dt \;=\; -\int_{t_n}^{t_{n+1}} A \int_{x_{j-1/2}}^{x_{j+1/2}} q_x(x\,,\,t)\,dx\,dt. \qquad (8a)$$

Dividing out the constant cross-sectional area A and integrating the derivatives yields

$$\int_{x_{j-1/2}}^{x_{j+1/2}} E(x\,,\,t)\,dx \,\Big|_{t=t_n}^{t=t_{n+1}} \;=\; \int_{t_n}^{t_{n+1}} [\,q(x_{j-1/2}\,,t) - q(x_{j+1/2}\,,t)\,]\,dt\;. \qquad (8b)$$

Assuming V_j is small enough for $E(x_j, t)$ to be approximately the mean energy (density) inside V_j, i.e. assuming E is approximately uniform in V_j we have

$$\int_{x_{j-1/2}}^{x_{j+1/2}} E(x\,,\,t)\,dx \;\approx\; E(x_j\,,\,t)\,\Delta x_j\;,$$

and (8) becomes

$$[\,E(x_j\,,\,t_{n+1}) - E(x_j\,,\,t_n)\,]\;\;\Delta x_j \;=\; \int_{t_n}^{t_{n+1}} [\,q(x_{j-1/2}\,,\,t) - q(x_{j+1/2}\,,\,t)\,]\,dt\;. \qquad (9)$$

This simply expresses the heat balance in V_j during $(\,t_n\,,\,t_{n+1}\,)$, namely, the gain of heat during this time is equal to the amount of heat entering the volume (from the left), minus the heat leaving it (at the right, per unit cross-sectional area).

Next, we assume that the time-increment Δt_n may be so brief that during the time $(\,t_n\,,\,t_{n+1}\,)$ the fluxes are approximately constant and arbitrarily close to their values at any intermediate time in this interval. Let

$$t_{n+\theta} \;:=\; t_n + \theta\Delta t_n \;=\; (1-\theta)t_n + \theta t_{n+1}\;, \qquad (10)$$

be some intermediate time with $0 \le \theta \le 1$. The usual choices are $\theta = 0,\ 1/2$ or 1, and these will be discussed later. We can then approximate (9) by

$$[\,E(x_j\,,\,t_{n+1}) - E(x_j\,,\,t_n)\,]\,\Delta x_j \;=\; \Delta t_n\,[\,q(x_{j-1/2}\,,\,t_{n+\theta}) - q(x_{j+1/2}\,,\,t_{n+\theta})\,]\,, \quad j = 1, \ldots, M\,, \tag{11}$$

which constitutes a *complete discretization of the conservation law* (5). To obtain a numerical scheme, we introduce the discrete approximations

$$E_j^n \;\approx\; E(x_j, t_n)\,, \qquad q_{j\pm 1/2}^{n+\theta} \;\approx\; q(x_{j\pm 1/2}\,,\; t_n + \theta\Delta t_n), \qquad 0 \le \theta \le 1\,,$$

and write (11) as

$$E_j^{n+1} - E_j^n \;=\; \frac{\Delta t_n}{\Delta x_j}\,[\,q_{j-1/2}^{n+\theta} - q_{j+1/2}^{n+\theta}], \qquad j = 1, \ldots, M, \qquad n = 0, 1, \ldots \tag{12}$$

This is the discretization of the energy conservation law that will be extended to phase change processes as the ''enthalpy method'' in **§4.3**. The thermal state of the control volume V_j at the time t_n is completely determined by the enthalpy E_j^n. Relation (12) provides us with the means for updating that thermal state to the next discrete time t_{n+1}.

Let us now discuss the choice of the parameter θ. For $\theta = 0$ the fluxes are evaluated at the old time, t_n, and (12) constitutes an **explicit** determination of the enthalpy approximation E^{n+1} of E at the advanced time step t_{n+1} in terms of the state of the material at t_n :

$$\textbf{explicit scheme}: \quad E_j^{n+1} \;=\; E_j^n + \frac{\Delta t_n}{\Delta x_j}\,[\,q_{j-1/2}^n - q_{j+1/2}^n]\,, \quad j = 1, \ldots, M\,, \quad n = 0, 1, \ldots\,. \tag{13}$$

For $\theta = 1$ we have the

$$\textbf{fully implicit scheme}: \quad E_j^{n+1} \;=\; E_j^n + \frac{\Delta t_n}{\Delta x_j}\,[\,q_{j-1/2}^{n+1} - q_{j+1/2}^{n+1}]\,, \quad j = 1, \ldots, M, \quad n = 0, 1, \ldots\,. \tag{14}$$

For $0 < \theta < 1$, the scheme is also implicit, using intermediate values of the flux which we define as

$$q^{n+\theta} \;:=\; (1-\theta)\,q^n + \theta\,q^{n+1}.$$

The most common choice is $\theta = 1/2$, in which case the resulting numerical method is known as the **Crank – Nicolson scheme**, about which more will be said later.

Discrete fluxes

These schemes require approximations of the fluxes across the faces located at $x_{j-1/2}$ and $x_{j+1/2}$. Let us consider *interior* control-volume faces first; the boundary cases will be discussed in **§4.1.C**. The conductive flux is given by

$$q \;=\; -kT_x \;\approx\; -k\,\frac{\Delta T}{\Delta x}\,. \tag{15}$$

Since the temperature is represented discretely by nodal values T_j, we may use first-order finite differences to approximate q discretely. Thus,

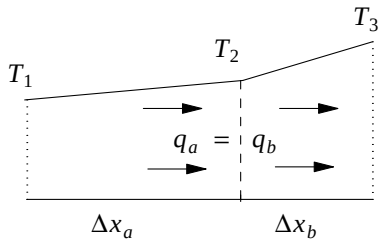

Figure 4.1.4. Steady-state profile.

$$q_{j-1/2} = -k_{j-1/2}\frac{T_j - T_{j-1}}{x_j - x_{j-1}}, \qquad j = 2, \ldots, M\ , \tag{16}$$

is the amount of heat flowing from V_{j-1} into V_j across a unit cross-sectional area per unit time. But what does $k_{j-1/2}$ represent? Generally, the conductivity is *not* constant but a function of *location* (when V_{j-1}, V_j consist of different materials, e.g. a wall and a phase change material or liquid and solid phases of the same material), and of *temperature*, $k = k(x, T)$. So, in general, the flux must represent heat flow through media of different conductivities, k_{j-1} for V_{j-1} and k_j for V_j, and we need to assign, in a consistent manner, an **effective conductivity** $k_{j-1/2}$. A reasonable definition of effective conductivity for a *layered* structure is obtained as follows.

Consider steady-state heat conduction ($T_{xx} = 0$) through two adjacent layers of thicknesses $\Delta x_a, \Delta x_b$ and conductivities k_a, k_b. Then the temperature profiles are straight lines (**Figure 4.1.4**) and at the common wall the flux from the left must equal the flux from the right:

$$-k_a\frac{T_2 - T_1}{\Delta x_a} = q = -k_b\frac{T_3 - T_2}{\Delta x_b}\,.$$

Solving the first equality for $T_2 - T_1$, the second for $T_3 - T_2$ and adding we obtain

$$T_3 - T_1 = -q\,(\frac{\Delta x_a}{k_a} + \frac{\Delta x_b}{k_b}).$$

Hence, the flux across the common wall is

$$q = -\frac{T_3 - T_1}{\dfrac{\Delta x_a}{k_a} + \dfrac{\Delta x_b}{k_b}}\,.$$

We refer to the ratio of length to conductivity as the **thermal resistance**. Hence, the relationship between flux q and resistance R is $q = -\dfrac{\Delta T}{R}$ where the temperature drop ΔT is often referred to as the **thermal driving force.**

It should be noted that the common definition of thermal resistance is

$$\frac{\textit{length of resistance path}}{(\textit{crosssectional area})\,(\textit{conductivity})},$$

making the formula

$$\text{heat flow rate} \;=\; qA \;=\; -k\,\frac{\Delta T}{\Delta x}\,A \;=\; -\frac{\Delta T}{\Delta x\,/\,Ak} \;=\; -\frac{\Delta T}{R}$$

correct. However, when the cross sectional area A is constant and $\Delta V = A\Delta x$, the A divides out in the discretization of the conservation law:

$$\Delta E \;=\; \frac{\Delta t}{\Delta V}\,[\![qA]\!]_{+}^{-} \;=\; \frac{\Delta t}{A\Delta x}\,[\![q]\!]_{+}^{-}A \;=\; \frac{\Delta t}{\Delta x}\,[\![q]\!]_{+}^{-}\,,$$

so we only need an expression for the flux and not the flow rate. Hence, for 1-dimensional Cartesian geometry, it is more convenient to take as resistance the quantity $\dfrac{\Delta x}{k}$ instead of the standard $\dfrac{\Delta x}{Ak}$ (See also **§4.1.F**).

From the above analysis we see that the effective overall resistance of the composite layer is $R = R_a + R_b$. Hence it is **not** the conductivities that add up but the resistances of the two layers, in this *serial* arrangement. We conclude that the total flux through a composite layer equals the overall temperature drop divided by the sum of the resistances of the layers.

With this in mind, we set (**Figure 4.1.5**)

$$R_{j-1/2} \;=\; \frac{\frac{1}{2}\Delta x_{j-1}}{k_{j-1}} + \frac{\frac{1}{2}\Delta x_j}{k_j} \;=\; \text{resistance of the path } [x_{j-1},\, x_j] \tag{17}$$

and express the interior fluxes, (16), as

$$q_{j-1/2} \;=\; -\frac{T_j - T_{j-1}}{R_{j-1/2}}\,, \qquad j = 2, \ldots, M \;\; . \tag{18}$$

In particular, if $\Delta x_{j\pm1} = \Delta x_j \;=\; \Delta x$ and $k_{j\pm1} = k_j = k$ then their common resistance is

$$R \;=\; \frac{\Delta x}{2}\,(\frac{1}{k} + \frac{1}{k}) \;=\; \frac{\Delta x}{k}\,,$$

as expected.

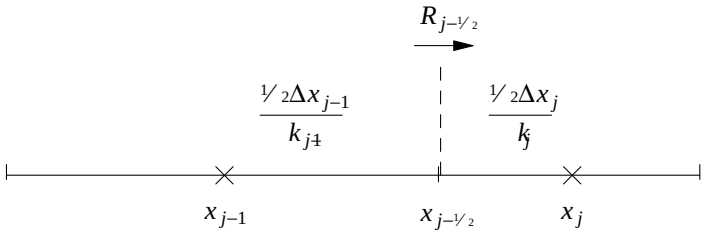

Figure 4.1.5. Resistances of adjacent control volumes.

Discrete heat equation

Substituting the flux expression (18) into the discrete heat balance (12) we obtain

$$E_j^{n+1} = E_j^n + \frac{\Delta t_n}{\Delta x_j}\left[\frac{T_{j+1}^{n+\theta} - T_j^{n+\theta}}{R_{j+½}} - \frac{T_j^{n+\theta} - T_{j-1}^{n+\theta}}{R_{j-½}}\right]$$

$$= E_j^n + \frac{\Delta t_n}{\Delta x_j}\left[\frac{1}{R_{j-½}} T_{j-1}^{n+\theta} - \left(\frac{1}{R_{j-½}} + \frac{1}{R_{j+½}}\right) T_j^{n+\theta} + \frac{1}{R_{j+½}} T_{j+1}^{n+\theta}\right]. \quad (19)$$

In particular, if $\Delta x_j = \Delta x$, and $k_j = k$, then $R_{j\pm½} = \dfrac{\Delta x}{k}$ and (19) becomes

$$E_j^{n+1} = E_j^n + \frac{k\Delta t_n}{\Delta x^2}\left[T_{j-1}^{n+\theta} - 2T_j^{n+\theta} + T_{j+1}^{n+\theta}\right]; \quad (20)$$

the bracketed expression is, of course, the standard centered finite-difference discretization of T_{xx}.

For *plain heat conduction*, the energy is simply the sensible heat, (6), which, when the specific heat is independent of temperature, becomes

$$E_j^n \approx E(x_j, t_n) = \rho c_j\left[T_j^n - T_{ref}\right] .$$

This can be used to eliminate E_j^n, and thus (12) takes the form

$$T_j^{n+1} = T_j^n + \frac{\Delta t_n}{\rho c_j \Delta x_j}\left[q_{j-½}^{n+\theta} - q_{j+½}^{n+\theta}\right], \qquad j = 1, \ldots, M, \qquad n = 0, 1, 2, \ldots \quad (21)$$

with $0 \le \theta \le 1$ to be chosen. This is often a convenient discretization, the fluxes being given by (18) for interior faces, and as described in **§4.1.C** for the boundary faces. Alternatively, the fluxes may be eliminated completely, using (18), to obtain

$$T_j^{n+1} = T_j^n + \frac{\Delta t_n}{\rho c_j \Delta x_j}\left[\frac{1}{R_{j-½}} T_{j-1}^{n+\theta} - \left(\frac{1}{R_{j-½}} + \frac{1}{R_{j+½}}\right) T_j^{n+\theta} + \frac{1}{R_{j+½}} T_{j+1}^{n+\theta}\right], \quad (22)$$
$$j = 1, \ldots, M; \quad n = 0, 1, 2, \ldots .$$

which is a complete discretization of the heat conduction equation (1) in terms of temperatures only, with $T_0^{n+\theta}$ and $T_{M+1}^{n+\theta}$ determined by the boundary conditions.

In particular, for a uniform grid, $\Delta x_j = \Delta x$, uniform time steps, $\Delta t_n = \Delta t$ and constant thermophysical properties ($\alpha = k/\rho c$) we have

$$T_j^{n+1} = T_j^n + \frac{\alpha\Delta t}{\Delta x^2}\left[T_{j-1}^{n+\theta} - 2T_j^{n+\theta} + T_{j+1}^{n+\theta}\right], \quad j = 1, \ldots, M; \quad n = 0, 1, 2, \ldots . \quad (23)$$

For $\theta = 0$, this is the usual explicit discretization of the heat equation, $T_t = \alpha T_{xx}$, obtained by forward Euler discretization of T_t and centered differencing of T_{xx}. For any $0 < \theta \le 1$, the discretization is implicit. Their pros and cons are discussed in **§4.1.E** and **§4.1.F**.

4.1.C Discretization of boundary conditions

For equations (12) (or (19) or (22)) to constitute a closed system allowing the state of the system to be advanced from t_n to t_{n+1}, values are needed for the boundary fluxes $q_{1/2}$ and $q_{M+1/2}$, representing the fluxes through the walls $x = 0$ and $x = l$ (or, values for T_0 and T_{M+1}). We discuss the treatment of boundary conditions at $x = 0$, the treatment at $x = l$ being completely analogous. The concept of thermal resistance makes the treatment of boundary conditions simple.

Case I. Imposed temperature: $T(0, t) = T_0(t)$

Here the wall temperature is specified, so the value of T_0^n is known at each time t_n,

$$T_0^n = T_0(t_n), \qquad n = 0, 1, 2, \ldots \tag{24}$$

Then from (18), the boundary flux is

$$q_{1/2}^n = -\frac{T_1^n - T_0^n}{R_{1/2}}, \quad \text{with} \quad R_{1/2} = \frac{1/2\Delta x_1}{k_1}. \tag{25}$$

Case II. Imposed Flux: $-kT_x(0, t) = q_0(t)$

Here the boundary flux is specified, so

$$q_{1/2}^n = q_0(t_n), \qquad n = 0, 1, 2, \ldots. \tag{26}$$

Then, the surface temperature T_0^n is obtained from $-\dfrac{T_1^n - T_0^n}{R_{1/2}} = q_0(t_n)$, whence

$$T_0^n = T_1^n + R_{1/2}q_0(t_n), \qquad \text{with} \quad R_{1/2} = \frac{1/2\Delta x_1}{k_1}. \tag{27}$$

Case III. Convective Flux: $-kT_x(0, t) = h[T_\infty(t) - T(0, t)]$

Setting $T_\infty^n := T_\infty(t_n)$, and employing the standard discretization

$$q_{1/2}^n = -\frac{T_1^n - T_0^n}{R_{1/2}}, \qquad R_{1/2} = \frac{1/2\Delta x_1}{k_1}, \tag{28}$$

of the conductive flux $-kT_x(0, t)$, we see that the boundary condition requires $-\dfrac{T_1^n - T_0^n}{R_{1/2}} = h[T_\infty^n - T_0^n]$, from which T_0^n is expressed as a weighted average of T_∞^n and T_1^n :

$$T_0^n = \frac{T_1^n + hR_{1/2}T_\infty^n}{1 + hR_{1/2}}. \tag{29}$$

Substituting this value of T_0^n into (28), we find

$$q^n_{1/2} = -\frac{T^n_1 - T^n_\infty}{1/h + R_{1/2}}. \tag{30}$$

Comparison with (28) reveals that the ambient temperature, T^n_∞, can play the role of the face temperature T^n_0 provided the conductive resistance, $R_{1/2}$, is replaced by the total effective resistance $\frac{1}{h} + R_{1/2}$, (the sum of the convective and conductive resistances).

As usual, the *imposed* temperature case, (25), corresponds to $h \to \infty$ in (29), (30).

4.1.D The discrete problem

Having derived discretizations of both the partial differential equation and the boundary conditions, we can now present algorithms for finding the unknown nodal temperatures $T^{n+1}_1, T^{n+1}_2, \cdots, T^{n+1}_M$, at time t_{n+1}, from their values at the old time t_n for our model heat conduction problem (1). Indeed, combining (4), (30), (26) but applied to $x = l$, and (21), (18), (17), the updating equations for the T^{n+1}_j's are as follows:

initial values: $$T^0_j = T_{init}(x_j), \quad j = 1, \ldots, M, \tag{31a}$$

boundary condition at $x = 0$: $$q^{n+\theta}_{1/2} = -\frac{T^{n+\theta}_1 - T^{n+\theta}_\infty}{\frac{1}{h} + R_{1/2}}, \quad R_{1/2} = \frac{1/2\Delta x}{k}, \tag{31b}$$

boundary condition at $x = l$: $$q^{n+\theta}_{M+1/2} = 0, \tag{31c}$$

interior values: $$T^{n+1}_j = T^n_j + \frac{\Delta t_n}{\rho c_j \Delta x_j}\left[q^{n+\theta}_{j-1/2} - q^{n+\theta}_{j+1/2} \right], \quad j = 1, \ldots, M, \tag{31d}$$

where

$$q^{n+\theta}_{j-1/2} = -\frac{T^{n+\theta}_j - T^{n+\theta}_{j-1}}{R_{j-1/2}} \quad \text{with} \quad R_{j-1/2} = \frac{1/2\Delta x_{j-1}}{k_{j-1}} + \frac{1/2\Delta x_j}{k_j}, \quad j = 2, \ldots, M, \tag{31e}$$

and

$$T^{n+\theta} = (1-\theta)T^n + \theta T^{n+1}, \qquad 0 \le \theta \le 1. \tag{31f}$$

The solvability of this system for the choices $\theta = 0$, $0 < \theta \le 1$, is discussed in the following subsections.

Note that neither the spatial steps Δx_j nor the time steps Δt_n need be uniform. A finer mesh may be needed near boundaries, or wherever steep gradients are expected, to resolve rapid variations, etc. However, unless there is specific reason to use non-uniform spatial grids, uniform ones are preferred because they are

simpler and they yield better accuracy. Moreover, if the heat transfer coefficient is not constant but a function of t, $T_\infty(t)$ and $T(0,t)$ as in the radiation boundary condition (**§1.2**), then in (31b) h is actually

$$h^{n+\theta} = h(t_{n+\theta}, T_\infty(t_{n+\theta}), T_0^{n+\theta}), \tag{32a}$$

where, by (29),

$$T_0^{n+\theta} = \frac{T_1^{n+\theta} + \dfrac{h^{n+\theta}\Delta x_1}{2k_1} T_\infty^{n+\theta}}{1 + \dfrac{h^{n+\theta}\Delta x_1}{2k_1}}, \tag{32b}$$

making the system highly nonlinear if $\theta > 0$. Similarly, if the specific heat and/or conductivity are functions of location and temperature, then c_j, and k_j actually change with time because of the temperature change, and must be evaluated at $t = t_{n+\theta}$. The resulting nonlinear system must be solved by some iterative method (see **§4.1.F**).

Discretization replaces a Partial Differential Equation, **PDE**$[u] = 0$, by a Finite-Difference Equation, **FDE**$[U_j^n] = 0$. The amount by which the exact solution u of the **PDE** fails to satisfy the **FDE** is called the

local truncation error: $\mathbf{te}_j^n := \mathbf{FDE}[u(x_j, t_m)]$.

Since **PDE**$[u(x_j, t_n)] = 0$, the truncation error may be viewed as the difference between **FDE** and **PDE** applied to $u(x_j, t_n)$. The discretization is **consistent** if $\mathbf{te}_j^n \to 0$ as $\Delta x, \Delta t \to 0$, which signifies that the **FDE** is indeed an approximation to the given **PDE** (instead of to some other PDE); see PROBLEM 14. On the other hand, the distance between the continuous and discrete solutions is measured by the

local discretization error: $\mathbf{de}_j^n := U_j^n - u(x_j, t_n)$.

The method is **convergent** if $\mathbf{de}_j^n \to 0$ as $\Delta x, \Delta t \to 0$, which signifies that the discrete solution does indeed approximate the exact solution, see PROBLEM 15. Note that U_j^n denotes the *exact* solution of the **FDE**. The actual *computed* solution $\tilde{U}_j^n$, however, may be contaminated by **roundoff errors** $\mathbf{re}_j^n = \tilde{U}_j^n - U_j^n$. These may be introduced at any point in the computation (because of unavoidable rounding of data or computed values). Such errors then propagate to subsequent time-steps and to neighboring points. Even though a single rounding error is typically negligibly small, the concern is that it may grow so fast as it propagates that substantial accuracy in the computed solution is lost (see **§4.1.E** and PROBLEM 16). The best we can hope for is that the numerical scheme does not amplify errors so that they grow faster than the exact solution of the **FDE**. In particular, if the exact solution does not grow, then errors should not be amplified. In this case the numerical method is called **stable**. Clearly, the actual (local) error in the numerical solution is the sum $\mathbf{de}_j^n + \mathbf{re}_j^n = \tilde{U}_j^n - u(x_j, t_n)$. In a convergent method, we can reduce $\mathbf{de}_j^n$ by taking smaller $\Delta x, \Delta t$ but then $\mathbf{re}_j^n$ increases

(see PROBLEM 21); hence, in practice, there is always an error in the computed results.

For any $0 \le \theta \le 1$, (31) is a consistent scheme with $\mathbf{te}_j^n = O(\Delta t + \Delta x^2)$, see PROBLEM 14. Stability is discussed in **§4.1.E, F** and convergence follows from the ([ISAACSON-KELLER], [LAPIDUS-PINDER])

Lax Equivalence Theorem: A *consistent* finite-difference method for a *well-posed* (linear) problem is *convergent* if and only if it is *stable*.

Also see PROBLEM 15.

4.1.E Explicit time updating

Choosing $\theta = 0$ in (31), the fluxes are evaluated at the old time t_n and therefore they are completely known. This amounts to assuming that the values of the fluxes do **not** change appreciably during the time interval $[t_n\ ,\ t_{n+1}]$, so that the process at time t_{n+1} is still driven by the fluxes at time t_n. The time discretization is then the standard forward Euler discretization, and the new values, T_j^{n+1}, are obtained directly, simply by evaluating the right-hand sides.

Written in terms of temperatures only, the explicit scheme consists of (PROBLEM 13)

$$T_j^0 \ = \ T_{init}(x_j), \qquad j = 1, \ldots, M\ , \tag{33a}$$

$$T_0^n \ = \ \frac{T_1^n + h\, R_{1/2} T_\infty^n}{1 + h\, R_{1/2}}, \qquad \text{where} \qquad R_{1/2} = \frac{\Delta x_1}{2k_1}, \tag{33b}$$

$$T_{M+1}^n \ = \ T_M^n - 0 \cdot R_{M+1/2}, \qquad \text{where} \qquad R_{M+1/2} \ = \frac{\Delta x_M}{2k_M} \tag{33c}$$

(since $q_{M+1/2} = 0$ in our example problem), and

$$T_j^{n+1} = T_j^n + \frac{\Delta t_n}{\rho c_j \Delta x_j}\left[\frac{1}{R_{j-1/2}} T_{j-1}^n - \left(\frac{1}{R_{j-1/2}} + \frac{1}{R_{j+1/2}}\right) T_j^n + \frac{1}{R_{j+1/2}} T_{j+1}^n\right], \quad j = 1, \ldots, M \tag{33d}$$

with

$$R_{j+1/2} \ = \ \frac{\Delta x_j}{2k_j} + \frac{\Delta x_{j+1}}{2k_{j+1}}, \qquad j = 1, 2, 3, \ldots, M-1\ , \text{ and}$$

$$R_{j-1/2} \ = \ \frac{\Delta x_{j-1}}{2k_{j-1}} + \frac{\Delta x_j}{2k_j}, \qquad j = 2, 3, \ldots, M\ . \tag{33e}$$

The **local truncation error** is of order Δt in time and Δx^2 in space ([SMITH], [LAPIDUS-PINDER], [SEWELL], PROBLEM 14). Clearly, if the thermal conductivity and specific heat are constants, $k_j \equiv k$, $c_j \equiv c$, and the mesh is uniform, $\Delta x_j \equiv \Delta x$, then $R_{j+1/2} = R_{j-1/2} \equiv \Delta x/k$, so setting

$$\mu \ = \ \frac{\alpha \Delta t}{\Delta x^2}, \qquad \alpha \ = \ \frac{k}{\rho c}\ , \tag{34}$$

we see that (33d) simplifies to (c.f. (23))

$$T_j^{n+1} = T_j^n + \mu [T_{j-1}^n - 2T_j^n + T_{j+1}^n], \qquad j = 2, \ldots, M , \tag{35}$$

for the internal nodes ($j = M$ is also included here since $T_{M+1}^n = T_M^n$ by (33c)). Boundary nodes are discussed later.

The extreme simplicity and convenience of the explicit scheme however is partially offset by the necessity of restricting the time step size to ensure the *numerical stability* of the scheme. This is easiest to explain in the simplest case of (35), which may be re-written as

$$T_j^{n+1} = (1-2\mu) T_j^n + \mu (T_{j-1}^n + T_{j+1}^n), \qquad j = 2, \ldots, M . \tag{36}$$

The condition for stability is that $1 - 2\mu \geq 0$, known as the

Courant – Friedrichs – Lewy (*CFL*) **Condition**: $$\Delta t \leq \frac{1}{2} \frac{\Delta x^2}{\alpha} , \tag{37}$$

after its discoverers [COURANT-FRIEDRICHS-LEWY]. It guarantees that the exact solution of the numerical scheme will obey a *Maximum Principle*, as does the solution of the Heat Equation itself, namely that

$$\min \{T_{j-1}^n, T_j^n, T_{j+1}^n\} \leq T_j^{n+1} \leq \max \{T_{j-1}^n, T_j^n, T_{j+1}^n\} .$$

Indeed, if condition (37) is violated, then (36) can produce physically unrealistic values, for example, a negative T_j^{n+1} from positive T_{j-1}^n, T_j^n, T_{j+1}^n. The numerical consequence is that errors would grow exponentially with n. Indeed, if errors are introduced at any time, from whatever source (say, roundoff), then (36) will compute contaminated values, $\tilde{\tilde{T}}_j^n = T_j^n + e_j^n$, instead of the desired values T_j^n in later steps; since both the T_j^n's and the $\tilde{\tilde{T}}_j^n$'s satisfy (36), the errors e_j^n also do:

$$e_j^{n+1} = (1 - 2\mu) e_j^n + \mu [e_{j-1}^n + e_{j+1}^n] , \qquad j = 2, \ldots, M ; \tag{38}$$

as a simple illustration, assuming $e_j^0 = (-1)^j e$, we find

$$e_j^1 = (1 - 4\mu) e_j^0, \qquad e_j^2 = (1 - 4\mu)^2 e_j^0, \ldots, \qquad e_j^n = (1 - 4\mu)^n e_j^0 ,$$

whence the error amplification factor is $1 - 4\mu$ at each step; it follows that, unless $|1 - 4\mu| \leq 1$, i.e. $0 \leq \mu \leq \frac{1}{2}$, the errors will be amplified exponentially fast and will destroy the computation after a few steps ! On the other hand, the requirement that Δt be greater than the relaxation time $\tau = \Delta x^2/\pi^2\alpha$ (PROBLEM 11) provides a lower bound $\mu > 1/\pi^2$.

More generally, in the **von Neumann stability analysis** approach, errors are represented by Fourier expansions and amplification factors of typical Fourier terms are determined, which leads to (37) as the condition for no-growth in the propagated errors, see [ALLEN-HERRERA-PINDER, p. 86, p. 206], [LAPIDUS-PINDER, p. 170]. Another approach to numerical stability is the ‘‘matrix

method'' [LAPIDUS-PINDER, p. 179].

A simple and effective way to guarantee stability is the ''*positive-coeffi cient rule*'': when T_j^{n+1} is written as a linear combination of its neighbors T_{j-1}^n, T_j^n, T_{j+1}^n (see (33d)), the coeffi cients must all be positive; this has been shown to be suffi cient for stability by [FORSYTHE-WASOW], (see [PATANKAR] for a discussion). Thus, in the more general case of (33d), *stability at internal nodes* is guaranteed by

$$\Delta t_n \le \min_{2\le j\le M-1} \frac{\rho c_j \Delta x_j}{\dfrac{1}{R_{j-\frac{1}{2}}} + \dfrac{1}{R_{j+\frac{1}{2}}}} \qquad \text{or simply} \qquad \Delta t_n \le \frac{1}{2}\frac{\Delta x_{min}^2}{\alpha_{max}}, \tag{39}$$

where $\Delta x_{min} = \min \Delta x_j$ and $\alpha_{max} = \max \alpha_j$.

We now consider boundary nodes for each type of boundary conditions. It is easy to see (PROBLEM 17) that if $T_0^n = T_0(t_n)$ is imposed at $x = 0$, then the coeffi cient of T_1^n in (33d) for $j = 1$ is $1 - \dfrac{\Delta t_n}{\rho c_1 \Delta x_1}\left(\dfrac{1}{R_{\frac{1}{2}}} + \dfrac{1}{R_{1+\frac{1}{2}}}\right)$, whose positivity is guaranteed by choosing

$$\Delta t_n \le \frac{1}{3}\frac{\Delta x^2}{\alpha}. \tag{40a}$$

Note that this restricts the time-step even more than (39). On the contrary, if the flux $q_{\frac{1}{2}}^n = q_0(t_n)$ is prescribed at $x = 0$, then the coeffi cient of T_1^n is $1 - \dfrac{\Delta t_n}{\rho c_1 \Delta x_1 R_{1+\frac{1}{2}}}$, whence it suffi ces to choose

$$\Delta t_n \le \frac{\Delta x^2}{\alpha} \tag{40b}$$

which will automatically hold under (39). Finally, the convective boundary condition case leads to the restriction

$$\Delta t_n \le \frac{1 + h\dfrac{\Delta x}{2k}}{1 + 3h\dfrac{\Delta x}{2k}} \cdot \frac{\Delta x^2}{\alpha}, \tag{40c}$$

for which (40a) is suffi cient.

We see that it suffi ces to restrict the time-step according to (PROBLEM 18)

$$\Delta t_n < \frac{1}{3}\frac{\Delta x_{min}^2}{\alpha_{max}} \qquad \begin{array}{l}\text{for } \textit{imposed temperature} \\ \text{or } \textit{convective boundary conditions},\end{array} \tag{41}$$

or

$$\Delta t_n < \frac{1}{2}\frac{\Delta x_{min}^2}{\alpha_{max}} \qquad \text{for } \textit{imposed flux boundary conditions}. \tag{42}$$

We have replaced $\le$ with $<$ as a precaution, against roundoff.

Such restrictions on the time-step size may be rather severe, making computations with an explicit scheme expensive. When the material properties vary with temperature, Δt_n needs to be re-adjusted (re-computed) before a new

time-step is taken. In such a case it is good programming practice *not* to let Δt become smaller than a pre-set minimum; for if it does, no practical time-advancing will be observed and computation will be wasted; instead, halt the computation and carefully examine what has caused the time step size to become so small.

Note also that (41) may be significantly more restrictive than (42) which is all that is required at internal nodes. Fortunately, there is a simple way of avoiding (41) entirely: for the boundary node(s) only, use the fully implicit discretization. For example, if $T_0^n = T_0(t_n)$ is imposed at $x = 0$, (also, see PROBLEM 20), then the implicit discretization at the boundary node ($j = 1$, $\theta = 1$ in (31d), assuming uniform Δx and constant k for simplicity) is

$$(1 + 3\mu)T_1^{n+1} = T_1^n + \mu[\,2T_0^{n+1} + T_2^{n+1}]\,; \tag{43}$$

with Δt_n as in (42), we update the internal nodes T_2^{n+1}, $T_3^{n+1}, \ldots,$ explicitly, $T_0^{n+1} = T_0(t_{n+1})$ is given, and therefore we can find T_1^{n+1} from (43).

4.1.F Implicit time updating

Choosing the parameter θ to be greater than zero in (31) results in a system of simultaneous equations for the unknowns T_1^{n+1}, T_2^{n+1}, $\cdots$, T_M^{n+1}, which must be solved, usually by some iterative method. The advantage of implicit schemes over explicit ones is their possible unconditional stability dependent on the choice of θ. The price to be paid is having to solve a system of equations, instead of just evaluations; we shall discuss some commonly used methods below.

The common choices for the value of θ are $1/2$ and 1. Choosing $\theta = 1$, the fluxes are computed at the latest time, t_{n+1}, and the scheme is referred to as **fully implicit**. It results from the backward Euler time discretization and its local error is again of order Δt in time and Δx^2 in space. In this regard, the **Crank-Nicolson** scheme, resulting from taking $\theta = 1/2$ in (31) is preferable. The fluxes, $q^{n+1/2}$ at the mid-point of the time-interval $[\,t_n\,,\,t_{n+1}\,]$ are taken as the averages of the values at t_n and t_{n+1}. This amounts to employing a centered-difference formula for the time derivative, resulting in a local error of order Δt^2 in time and Δx^2 in space.

Written entirely in terms of temperatures, the implicit scheme for any $0 < \theta \le 1$ takes the form

$$-\frac{\theta\Delta t_n}{\rho c_j \Delta x_j}\frac{T_{j-1}^{n+1}}{R_{j-\frac12}} + \left[1 + \frac{\theta\Delta t_n}{\rho c_j \Delta x_j}\left(\frac{1}{R_{j-\frac12}} + \frac{1}{R_{j+\frac12}}\right)\right]T_j^{n+1} - \frac{\theta\Delta t_n}{\rho c_j \Delta x_j}\frac{T_{j+1}^{n+1}}{R_{j+\frac12}} \tag{44a}$$

$$= \frac{(1-\theta)\Delta t_n}{\rho c_j \Delta x_j}\frac{T_{j-1}^{n}}{R_{j-\frac12}} + \left[1 - \frac{(1-\theta)\Delta t_n}{\rho c_j \Delta x_j}\left(\frac{1}{R_{j-\frac12}} + \frac{1}{R_{j+\frac12}}\right)\right]T_j^{n} + \frac{(1-\theta)\Delta t_n}{\rho c_j \Delta x_j}\frac{T_{j+1}^{n}}{R_{j+\frac12}}\,,$$
$$j = 1, \ldots, M\,,$$

while the boundary conditions (31b, c) contribute the equations

$$T_0^{n+1} = \frac{T_1^{n+1} + hR_{1/2}T_\infty^{n+1}}{1 + hR_{1/}}, \qquad T_{M+1}^{n+1} = T_M^{n+1} - 0 \cdot R_{M+1/2} \ . \tag{44b}$$

These constitute a linear system of $M+2$ equations for the $M+2$ unknowns T_0^{n+1}, $T_1^{n+1}, \ldots, T_M^{n+1}, T_{M+1}^{n+1}$.

Let us examine this system in the simplest case of uniform $\Delta t_n = \Delta t$, $\Delta x_j = \Delta x$, and constant $c_j = c$ and $k_j = k$. Then, setting

$$\mu = \frac{\Delta t}{\rho c \Delta x} \frac{1}{R_{j\pm 1/2}} = \frac{k\Delta t}{\rho c \Delta x^2} = \frac{\alpha \, \Delta t}{\Delta x^2}, \tag{45}$$

the system consists of the $M+2$ equations

$$(1 + \frac{h\Delta x}{2k})T_0^{n+1} - T_1^{n+1} = \frac{h\Delta x}{2k} T_\infty^{n+1}, \tag{46a}$$

$$-\theta\mu T_{j-1}^{n+1} + (1 + 2\theta\mu)T_j^{n+1} - \theta\mu T_{j+1}^{n+1} = \tag{46b}$$
$$(1-\theta)\mu T_{j-1}^n + [1 - 2(1-\theta)\mu]T_j^n + (1-\theta)\mu T_{j+1}^n$$

$$-T_M^{n+1} + T_{M+1}^{n+1} = 0 \cdot \frac{\Delta x}{2k} \tag{46c}$$

The last equation simply says $T_{M+1}^{n+1} = T_M^{n+1}$, so we omit it, and write the remaining $M+1$ equations for the unknowns $T_0^{n+1}, T_1^{n+1}, \cdots, T_M^{n+1}$ in matrix form:

$$\begin{bmatrix} (1+\frac{h\Delta x}{2k}) & -1 & 0 & \cdots & \cdots & 0 \\ -\theta\mu & (1+2\theta\mu) & -\theta\mu & \cdots & \cdots & 0 \\ 0 & -\theta\mu & (1+2\theta\mu) & \cdots & \cdots & 0 \\ \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & -\theta\mu & (1+2\theta\mu) & -\theta\mu \\ 0 & \cdots & \cdots & 0 & -\theta\mu & (1+2\theta\mu) \end{bmatrix} \begin{bmatrix} T_0^{n+1} \\ T_1^{n+1} \\ \cdots \\ \cdots \\ T_{M-1}^{n+1} \\ T_M^{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{h\Delta x}{2k} T_\infty^{n+1} \\ (1-\theta)\mu T_0^n + [1-2(1-\theta)\mu]T_1^n + (1-\theta)\mu T_2^n \\ \cdots \\ \cdots \\ (1-\theta)\mu T_{M-2}^n + [1-2(1-\theta)\mu]T_{M-1}^n + (1-\theta)\mu T_M^n \\ (1-\theta)\mu T_{M-1}^n + [1-(1-\theta)\mu]T_M^n \end{bmatrix} \tag{47}$$

The coeffi cient matrix has several important properties. It is **tridiagonal** and **strictly diagonally dominant**, meaning that the magnitude of each diagonal entry

is greater than the sum of the absolute values of the off-diagonal entries, $|1 + 2\theta\mu| > |-\theta\mu| + |-\theta\mu|$. Moreover, multiplying the first equation by $\theta\mu$ makes the coefficient matrix symmetric. It is known, (see, for example, [SEWELL]) that such a matrix is positive-definite and the elements of its inverse are all positive. It follows that the linear system always has a unique solution, which may be obtained by the very efficient **tridiagonal algorithm** (a variant of Gaussian elimination, see [PRESS et al], [MINKOWYCZ et al], [SEWELL]).

In more general cases, the tridiagonal system (47) may be solved by the Gauss-Seidel iterative method or, more efficiently, by the SOR iterative method ([YOUNG-GREGORY], [LAPIDUS-PINDER], [SEWELL]). We shall discuss these later for phase change problems (see **§4.3**)

The advantage of implicit schemes lies in their improved stability properties. Indeed, the scheme (46) with $\frac{1}{2} \le \theta \le 1$ is known to be *unconditionally stable* [ISAACSON-KELLER], thus imposing no restriction on the time-step. Yet in practice, the Crank-Nicolson scheme ($\theta = \frac{1}{2}$) may exhibit oscillations for large time-steps (see [PATANKAR] for a discussion). In fact, the ''positive coefficient rule'' mentioned earlier, when applied to (46), requires $1 - 2(1-\theta)\mu \ge 0$, i.e.,

$$\mu \le \frac{1}{2(1-\theta)} , \tag{48}$$

which imposes a restriction on the time step for any $0 \le \theta < 1$. Only the fully implicit scheme ($\theta = 1$) is truly unconditionally stable in this stronger sense!

4.1.G Heat conduction in 2 or 3 dimensions

All of the previous developments generalize naturally to 2 or 3 space dimensions. We shall outline the treatment for a 3-dimensional analogue of the model heat conduction problem (1).

For simplicity, we consider a box, $\Omega : 0 \le x \le l_1$, $0 \le y \le l_2$, $0 \le z \le l_3$, initially at temperature $T_{init}(\vec{x})$. The face $x = 0$ is heated convectively, from a source at ambient temperature $T_\infty(t)$, the face $y = 0$ is kept at a fixed temperature T_{fixed} (for variety!) and the other faces are insulated (**Figure 4.1.6**). The parameters ρ , c , k will be assumed to be constant.

MATHEMATICAL PROBLEM : Find $T(\vec{x}, t) = T(x, y, z, t)$ such that

$$\rho c T_t = \nabla \cdot (k \nabla T) \qquad \text{in } \Omega , \quad t > 0 , \tag{49a}$$

$$T(\vec{x}, 0) = T_{init}(\vec{x}), \qquad \vec{x} \in \Omega , \tag{49b}$$

$$-kT_x \Big|_{x=0} = h\,[\,T_\infty(t) - T(0, y, z, t)\,] , \qquad T(x, 0, z, t) = T_{fixed} , \tag{49c}$$

$$-kT_x \Big|_{x=l_1} = -kT_y \Big|_{y=l_2} = -kT_z \Big|_{z=0} = -kT_z \Big|_{z=l_3} = 0 . \tag{49d}$$

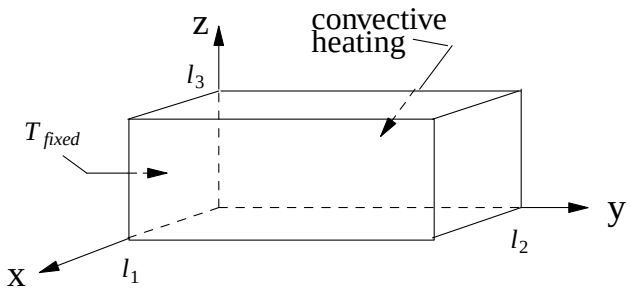

Figure 4.1.6. Melting of a box.

We subdivide $[0\,,\,l_1]$ into M_1 subintervals, $[0\,,\,l_2]$ into M_2 subintervals and $[0\,,\,l_3]$ into M_3 subintervals. For simplicity we take uniform grids in each direction, so that

$$\Delta x \;=\; \frac{l_1}{M_1}\,, \qquad \Delta y = \frac{l_2}{M_2}\,, \qquad \Delta z = \frac{l_3}{M_3}\,,$$

and $\Delta V = \Delta x \Delta y \Delta z$. Thus, the box Ω is subdivided into $M_1 M_2 M_3$ boxes V_{ijk} of uniform volume ΔV with centers $(x_i\,,\,y_j\,,\,z_k)$ and bounding surface ∂V_{ijk}. Approximations to the temperature $T(x_i\,,\,y_j\,,\,z_k\,,\,t_n)$ will be denoted by T^n_{ijk}, and to the energy density $E(x_i\,,\,y_j\,,\,z_k\,,\,t_n)$ by E^n_{ijk}, considered as mean values over V_{ijk}. Integrating the conservation law

$$E_t + \nabla \cdot \vec{q} \;=\; 0 \tag{50}$$

over the control volume V_{ijk} and $[t_n\,,\,t_{n+1}]$, we obtain similarly to (6)-(12), the discrete conservation law

$$\begin{aligned} E^{\,n+1}_{ijk} - E^{\,n}_{ijk} \;&=\; -\frac{\Delta t}{\Delta V_{ijk}} \int\limits_{\partial V_{ijk}} \vec{q}^{\,n+\theta} \cdot \vec{N}\, dS \\ &=\; \frac{\Delta t}{\Delta V_{ijk}} \big[q^{\,n+\theta}_{i-1/2\;j\,k} \cdot A_{i-1/2\;j\,k} - q^{\,n+\theta}_{i+1/2\;j\,k} \cdot A_{i+1/2\;j\,k} + q^{\,n+\theta}_{i\;j-1/2\;k} \cdot A_{i\;j-1/2\;k} \\ &\qquad - q^{\,n+\theta}_{i\;j+1/2\;k} \cdot A_{i\;j+1/2\;k} + q^{\,n+\theta}_{i\;j\;k-1/2} \cdot A_{i\;j\;k-1/2} - q^{\,n+\theta}_{i\;j\;k+1/2} \cdot A_{i\;j\;k+1/2} \big] \end{aligned} \tag{51}$$

$$\text{for} \quad i = 1, \ldots, M_1\,, \quad j = 1, \ldots, M_2\,, \quad k = 1, \ldots, M_3 \quad \text{and} \quad 0 \le \theta \le 1\,,$$

the A's denoting the areas of the corresponding faces.

In the rectangular geometry chosen, the areas of pairs of opposite faces are the same and $\Delta V = \Delta x \Delta y \Delta z$, so (51) simplifies to

$$\begin{aligned} E^{\,n+\theta}_{ijk} \;=\; E^{\,n}_{ijk} &+ \frac{\Delta t}{\Delta x} \big[q^{\,n+\theta}_{i-1/2\;j\,k} - q^{\,n+\theta}_{i+1/2\;j\,k} \big] + \frac{\Delta t}{\Delta y} \big[q^{\,n+\theta}_{i\;j-1/2\;k} - q^{\,n+\theta}_{i\;j+1/2\;k} \big] \\ &+ \frac{\Delta t}{\Delta z} \big[q^{\,n+\theta}_{i\;j\;k-1/2} - q^{\,n+\theta}_{i\;j\;k+1/2} \big]\,. \end{aligned} \tag{52}$$

In general, however, some pairs of opposite faces may have different areas (e.g. in the radial direction for the case of cylindrical geometry) and the above simplification will be unfeasible. In such a case, we need to express the heat flow rate, $q \cdot A$ and not just the flux (**§4.1.B**) in terms of temperature gradients, and use of the standard resistance becomes more convenient. So, in general, we define

$$\tilde{\tilde{R}}_{i-\frac{1}{2}\,j\,k} = \frac{1}{A_{i-\frac{1}{2}\,j\,k}}\left(\frac{\frac{1}{2}\Delta x_{i-1}}{k_{i-1\,j\,k}} + \frac{\frac{1}{2}\Delta x_i}{k_{i\,j\,k}}\right) \equiv \frac{1}{A_{i-\frac{1}{2}\,j\,k}} \cdot R_{i-\frac{1}{2}\,j\,k}\,, \tag{53}$$

and similarly for $\tilde{\tilde{R}}_{i+\frac{1}{2}\,j\,k}$, $\tilde{\tilde{R}}_{i\,j-\frac{1}{2}\,k}$, etc, so that the heat flow rate may be expressed as

$$(qA)^{n+\theta}_{i-\frac{1}{2}\,j\,k} = -\frac{T^{n+\theta}_{i\,j\,k} - T^{n+\theta}_{i-1\,j\,k}}{\tilde{\tilde{R}}_{i-\frac{1}{2}\,j\,k}}\,, \qquad \text{etc}\ . \tag{54}$$

In the simpler case of (52), we have

$$q^{n+\theta}_{i-\frac{1}{2}\,j\,k} = \frac{(qA)^{n+\theta}_{i-\frac{1}{2}\,j\,k}}{A_{i-\frac{1}{2}\,j\,k}} = -\frac{T^{n+\theta}_{ijk} - T^{n+\theta}_{i-1\,j\,k}}{\Delta y \Delta z \tilde{\tilde{R}}_{i-\frac{1}{2}\,j\,k}} = -\frac{T^{n+\theta}_{ijk} - T^{n+\theta}_{i-1\,j\,k}}{\frac{\Delta x}{2}\left(\frac{1}{k_{i-1\,j\,k}} + \frac{1}{k_{ijk}}\right)}\,, \quad \text{etc}\,. \tag{55}$$

The boundary conditions are discretized as in the 1-dimensional case (**§4.1.C**). For example, the convective flux boundary condition on the face $x = 0$ becomes

$$(qA)^{n+\theta}_{1-\frac{1}{2}\,j\,k} = -\frac{T^{n+\theta}_{1\,jk} - T^{n+\theta}_{\infty}}{\frac{1}{h\,A_{\frac{1}{2}\,j\,k}} + \tilde{\tilde{R}}_{\frac{1}{2}\,j\,k}} \equiv -\frac{T^{n+\theta}_{1\,jk} - T^{n+\theta}_{\infty}}{\left(\frac{1}{h} + R_{\frac{1}{2}\,j\,k}\right)\frac{1}{A_{\frac{1}{2}\,j\,k}}} \tag{56}$$

with $R_{\frac{1}{2}\,j\,k} = \frac{\frac{1}{2}\Delta x}{k_{1\,j\,k}}$; the imposed temperature on the face $y = 0$ becomes

$$(qA)^{n+\theta}_{i\,1-\frac{1}{2}\,k} = -\frac{T^{n+\theta}_{i1k} - T_{fixed}}{\tilde{\tilde{R}}_{i\,1-\frac{1}{2}\,k}} \equiv -\frac{T^{n+\theta}_{i1k} - T_{fixed}}{\frac{1}{A_{i\,\frac{1}{2}\,k}} R_{i\,\frac{1}{2}\,k}} \quad \text{with} \quad R_{i\,\frac{1}{2}\,k} = \frac{\frac{1}{2}\Delta y}{k_{i1k}}; \tag{57}$$

and the zero flux on the face $x = l_1$ becomes

$$(qA)^{n+\theta}_{M_1+\frac{1}{2}\,j\,k} = 0. \tag{58}$$

For the heat conduction process we are examining here, the energy is simply the sensible heat measured relative to some convenient T_{ref} :

$$E_{ijk} = \rho c_{ijk}[T_{ijk} - T_{ref}]. \tag{59}$$

When everything is expressed in terms of the temperatures, the equation for T^{n+1}_{ijk} involves the temperatures of the 6 adjacent nodes. Choosing $\theta = 0$ (explicit scheme), these neighboring temperatures will be at time t_n and thus known. The stability condition becomes

$$\Delta t_n \;<\; \frac{\min(\,\Delta x_i^2\,,\,\Delta y_j^2\,,\,\Delta z_k^2\,)}{6\cdot\max\alpha_{ijk}}\,, \tag{60}$$

often making computations lenghty and prohibitively expensive.

In the implicit case ($\theta = 1/2$ or 1), the resulting linear system will be hepta-diagonal and diagonally- dominant, so again it may be solved efficiently, especially if the ADI method is used to reduce the system to three tridiagonal ones, see [ALLEN-HERRERA-PINDER], [LAPIDUS-PINDER].

Cylindrical and spherical geometries are examined in PROBLEMS 5, 6, 25, 26, 28, 29.

4.1.H Internal heat source

The presence of an internal (volumetric) heat source adds a term in the energy conservation law (**§1.2**) , which, instead of (50) will read

$$E_t + \nabla\cdot\vec{q} \;=\; f \quad . \tag{61}$$

The source term $f(\vec{x},t)$ is the power density, representing the amount of energy delivered at location $\vec{x}$ at time t per unit volume per unit time (so it may be in units of $J/s\,cm^3 = Watts/cm^3$).

Its integration over V_{ijk} and $[t_n\,,\,t_{n+1}]$ contributes the additional term

$$\frac{1}{\Delta V_{ijk}}\int_{t_n}^{t_{n+1}}\int_{V_{ijk}} f(\vec{x}\,,\,t)\,dVdt \tag{62}$$

in (51). It should be noted that this integral should *not* be discretized by the low order approximations used for the derivative terms because large errors may ensue. The integration in (62) should be performed analytically whenever possible, or high order numerical integration methods should be employed. The result may be represented as $S(\Delta V_{ijk}\,,\,\Delta t_n)$, and its discrete approximation as S_{ijk}^n. With this term added, the numerical scheme remains the same in all other aspects. In particular, the stability condition, (60), for the explicit scheme is not altered.

In some processes the power density, f, may also depend on temperature, $f(\vec{x}\,,\,t\,,\,T(\vec{x},t))$, making its treatment difficult. Direct integration of (62) is now impossible and the low order discretization of T is imposed on this term as well. Its mean value approximation,

$$\frac{\Delta t_n}{\Delta V_{ijk}}\,\Delta V_{ijk}\,f(x_{ijk},t_{n+\theta}\,,\,T_{ijk}^{n+\theta})$$

may introduce large errors, unless the time step Δt_n is taken to be small. Some expedient ways for handling nonlinear source terms are suggested by [PATANKAR].

4.1.I Some programming suggestions

In implementing the schemes described above there are some steps that can be taken to enhance the utility, efficiency and maintainability of the code. Let us describe some points related to the construction, output, debugging and validation of a code, for the benefit of inexperienced programmers.

For clarity, readability, and adaptability, it is a good idea to place the control logic of the algorithm into the MAIN PROGRAM and code the various tasks as subroutines, e.g. INPUT, MESH, START, FLUX, PDE, OUTPUT, which will be called by MAIN. An example of such a structure is shown in **Table 4.1.1** below, as it would pertain to the explicit scheme applied to a slab with imposed temperatures at its ends. Comments and explanations of what is done are very helpful. In coding the algorithm, care should be taken to avoid unnecessary or inefficient computation. For example, expressions should be arranged so as to minimize loss of significant digits; polynomials should be evaluated in nested form; wherever an expression is used several times, evaluate it once and then use its value; ''if'' statements, and especially subroutine calls are relatively expensive, so their use should be minimized; frequent output slows down execution, so unnecessary output should be avoided. The longer the runs one plans to make, the more attention should be paid to such simple programming issues.

The computation begins by calling **Subroutine INPUT**, which reads in the data file, for example:

```
read    tmax, maxsteps, dtout
read    l, M            !   l = slab length, M = number of nodes
read    ρ, c, k         !   material properties
read    T_init, T_0, T_l !  initial and boundary temperatures

where   tmax      =   desired duration of the simulation,
        maxsteps  =   maximum number of time-steps to be allowed
                      for the entire computation,
        dtout     =   desired time-interval for output.
```

The time-stepping will be monitored both by the actual *time* and by the number of time-steps taken , *nsteps* (see **Table 4.1.1**), and will end when $time > tmax$ **or** $nsteps > maxsteps$, the latter as a precaution just in case the time-step dt gets to be too small (or even negative!) for any reason.

After the data have been read in, **Subroutine MESH** sets up the mesh structure (defines locations of nodes $x(i)$, control-volume faces, areas, etc.), and determines the appropriate time-step Δt. At each time step, time will be advanced by Δt.

In order to obtain output at precisely the desired time intervals, we introduce the variable *tout* = output time, which will be advanced only after each output step; before each *tout* is reached we temporarily reduce Δt, if necessary, so that $time + \Delta t$ equals *tout*; then Δt is restored to its permanent value. (see **Table 4.1.1**).

Table 4.1.1 Example of a driver code

```
c       HEAT.f : Heat conduction in a slab with imposed temperatures
c  7-17-91 : entered and debugged basic code
c------- Notation ----------
        (explanation of symbols used and their meaning)
c**************************
        Program HEAT
        (common blocks)
c---------- Initialize ----------
        call INPUT
        call MESH
          dtperm = dt
          tout = max(dtout, dt)
        time = 0.
        call START
c---------- Begin time-stepping --------
100     continue
        time = time + dt
        nsteps = nsteps + 1
        if( time .gt. tmax .OR. nsteps .gt. maxsteps ) go to 1000
        call FLUX
        call PDE
        if (time .eq. tout) then
          call OUTPUT
          dt = dtperm
          tout = tout + max(tout, dt)
        else if (time + dt .gt. tout) then
          dt = tout − time
        end if
        go to 100
c---------- end of time-stepping ----------
1000    continue
        (write any exiting information, such as time)
        stop
        end
```

Subroutine START initializes variables to their values at time = 0. Then each time-step consists of calling **FLUX**, which computes resistances and heat-flow-rates, and **PDE**, which solves the PDE.

Subroutine OUTPUT writes out the current values of the quantities of interest, such as temperature. It is usually desirable to have output at certain specified locations (where thermocouples may be located, for example), but it is generally overly complex to attempt to arrange the mesh in such a way that all these output locations coincide with computational nodes; instead, one may extract values at

the desired locations via interpolation of the nodal values.

To debug the code, one usually starts with short-time runs on a very coarse mesh, say with $M = 10$ nodes, and observes the behavior of the solution as various parameters are varied, watching out for any non-physical behavior, for example violation of the Maximum Principle. A crucial and necessary check is provided by an energy-balance check from time-step to time-step: the total energy of the system at time t_n is $E^n_{total} = \sum_{i=1}^{M} E^n_i \Delta V_i$, so the energy gain during $[t_n\ ,\ t_n + \Delta t_n]$, is $E^{n+1}_{total} - E^n_{total}$; this must equal the energy input from the boundaries, $\int_{t_n}^{t_n+\Delta t_n} \int_{\partial\Omega} \vec{q} \cdot \vec{N}\, dS$, i.e. the sum of the boundary flow-rates times Δt_n, (PROBLEM 12).

The final step in preparing a code is to validate it by running one or more *benchmark problems* on it with known solutions, and comparing the computed and exact solutions. One may start with a coarse mesh, and successively double the number of nodes (halving the mesh width Δx) to verify that various measures of the error (e.g. $\max_{1 \le j \le m} |T^n_j - T(x_j, t_n)|$ at a fixed t_n, or $\max_{0 \le t_n \le t_{max}} (\max_{1 \le j \le M} |T^n_j - T(x_j, t_n)|)$ decrease as M increases. For the algorithms described earlier, one expects to see errors of order $O(\Delta x^2)$, which is the order of the discretization error, at least for M's up to about 100 (in single precision); for larger M however, roundoff error takes over and the accuracy actually deteriorates, see PROBLEM 21.

Two- and three-dimensional simulations can easily tax the capabilities of even ''large'' mainframe computers, so vector or/and parallel ''super-computing'' becomes necessary. Then one must use various programming ''tricks'' to take advantage of the special features of such machines. For example, to aid vectorization one may unfold 2- or 3-dimensional arrays into 1-dimensional long vectors using ''red-black ordering'', and replace ''if'' statements with logical (boolean) equivalents inside DO loops, [WILLIAMS-WILSON], [ORTEGA-VOIGT]. Parallelism for transient problems may be achieved by ''domain-decomposition'' methods at a basic level [DRAKE-NARANG]. These new computing technologies, which are currently under intense development, have brought about a re-examination of the various serial algorithms to see which methods are best suited to the various machine architectures and classes of problems.

PROBLEMS

PROBLEM 1. Write a brief essay on the simulation of a thermal process, addressing the possible reasons for preparing it, the roles that it is to serve, the kinds of information available to it, the type of output it is to provide, the importance or lack of importance of computational speed and the accuracy that it is to

have. Specific points that you might address are simulations in various contexts, including laboratory scale studies, industrial size studies, and real time control.

PROBLEM 2. For the model heat conduction problem of **§4.1.A**, describe *qualitatively* how you expect the temperature to evolve in time. In particular, what is the expected appearance of temperature-time curves at preassigned thermocouple locations at the faces of the slab and at, say, two interior points? If a *fluxmeter* were attached at each of the faces, what flux-time curves would you expect to see?

PROBLEM 3. Set up a 1-dimensional *radial* mesh for *axially symmetric* heat transfer in a hollow *cylinder* $R_{in} \le r \le R_{out}$ of *unit height*. That is, subdivide the interval $[R_{in}, R_{out}]$ into M subintervals and determine the nodes r_i, faces $r_{i-1/2}$, radial "areas" $A_{i-1/2} = 2\pi r_{i-1/2}$, and control "volumes" $\Delta V_i = \pi r_{i+1/2}^2 - \pi r_{i-1/2}^2 = 2\pi r_i \Delta r_i$. For further developments see PROB. 9, 22.

PROBLEM 4. Set up a 1-dimensional *radial* mesh for *spherically symmetric* heat transfer in a solid *sphere* $0 \le r \le R_{out}$, by subdividing $[0, R_{out}]$ into M subintervals, [Here $A_{i-1/2} = 4\pi r_{i-1/2}^2$, $\Delta V_i = (4/3)\pi(r_{i+1/2}^3 - r_{i-1/2}^3)$]. See PROBLEMS 10, 23.

PROBLEM 5. Set up a 2-dimensional (r, z) mesh for *axially symmetric* heat transfer in a hollow *cylinder* $R_{in} \le r \le R_{out}$, $0 \le z \le Z$, by subdividing $[R_{in}, R_{out}]$ into M_r subintervals and $[0, Z]$ into M_Z subintervals. Determine the nodes (r_i, z_j), faces $r_{i-1/2}$, $z_{j-1/2}$, areas of radial faces $A_{i-1/2\,j}$, of axial faces $A_{i\,j-1/2}$, and control volumes ΔV_{ij}. See PROBLEMS 25, 28.

PROBLEM 6. Set up a 2-dimensional (r, θ) mesh for *axially symmetric* heat transfer in a *sphere* $0 \le r \le R_{out}$, by subdividing $[0, R_{out}]$ into M_r subintervals and $[0, \pi]$ into M_θ sectors (the right-half sphere suffices, so let $x = r \sin\theta$, $z = r \cos\theta$, with θ the azimouthal angle measured off the positive z-axis). Determine the nodes (r_i, θ_j), faces $r_{i-1/2}$, $\theta_{j-1/2}$, areas of radial faces $A_{i-1/2\,j} = 2\pi \int_{\theta_{j-1/2}}^{\theta_{j+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} x\, r\, dr\, d\theta$, of angular faces $A_{i\,j-1/2} = 2\pi \int_{r_{i-1/2}}^{r_{i+1/2}} x\, dr$, and control volumes $\Delta V_{ij} = 2\pi \int_{\theta_{j-1/2}}^{\theta_{j+1/2}} \int_{r_{i-1/2}}^{r_{j+1/2}} x\, r\, dr\, d\theta$.

[Check: $A_{i-1/2\,j} = 2\pi r_{i-1/2}^2(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})$, $A_{ij-1/2} = \pi(r_{i+1/2}^2 - r_{i-1/2}^2)\sin\theta_{j-1/2}$, $\Delta V_{ij} = (2\pi/3)(r_{i+1/2}^3 - r_{i-1/2}^3)(\cos\theta_{j-1/2} - \cos\theta_{j+1/2})$]. See PROBLEMS 26, 29.

PROBLEM 7. Discuss the factors that would lead you to use non-uniform spatial and time subdivisions. In particular, what would you do if results are desired at definite times (e.g. in accordance with the readings of some recording device), and under conditions where high temperature gradients are present in certain locations. Under what conditions would the latter actually occur?

PROBLEM 8. Using the Taylor expansion, show that for the centered-nodes mesh (2b) the error in the approximation of the mean value by the nodal value is $O(\Delta x_j^3)$.

PROBLEM 9. Derive the discrete heat balance, analogous to (11), or (12), for axially symmetric heat conduction in a cylinder of unit height, using the mesh constructed in PROBLEM 3.

PROBLEM 10. Derive the discrete heat balance for spherically symmetric heat conduction in a sphere, using the mesh of PROBLEM 4.

PROBLEM 11. The time increment Δt used in a time-stepping scheme should be so large that local equilibrium obtains in a control volume during this time, i.e. Δt should be larger than the **relaxation time** τ of the heat conduction process. This is the time required for the temperature to relax to its equilibrium (steady-state) value T_∞, relative to its initial distance from the steady state; the convenient and commonly used definition of the relaxation time τ is (e.g. see [PINSKY]): $\left| T(x, \tau) - T_\infty \,/\, T(x, 0) - T_\infty \right| = 1 / e$ or, equivalently,

$$1 / \tau := \lim_{t \to \infty} (1 / t) \ln | T(x, t) - T_\infty |.$$

Consider a control volume $0 \le x \le \Delta x$ of width Δx. Using the fact that the solution of the heat equation with vanishing boundary values (hence $T_\infty = 0$ here) is given by $T(x, t) = \exp(-\pi^2 \alpha\, t / \Delta x^2) \sin(\pi x / \Delta x)$, show that the relaxation time is $\tau = \Delta x^2 / \pi^2 \alpha$. Then, the requirement $\Delta t > \tau$ combined with the CFL condition restrict the ratio $\mu = \alpha \Delta t / \Delta x^2$ to be $1 / \pi^2 < \mu < 1 / 2$.

PROBLEM 12. Prove that with all choices of θ the numerical scheme (12) obeys a global heat balance identically.

PROBLEM 13. Choose $\theta = 0$ in (31) to derive the explicit scheme (33).

PROBLEM 14. For the simplest explicit scheme (35), we have $\mathbf{PDE}[T] \equiv T_t - \alpha T_{xx}$ and $\mathbf{FDE}[T_j^n] \equiv \dfrac{T_j^{n+1} - T_j^n}{\Delta t} - \alpha \dfrac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{\Delta x^2}$, see **§4.1.D**. Using Taylor expansions show that the *local truncation error* is given by

$$\mathbf{te}_j^n = \frac{\Delta t}{2} T_{tt}(x_j, t_n) - \alpha \frac{\Delta x^2}{12} T_{xxxx}(x_j, t_n) + O(\Delta t^2 + \Delta x^4) = O(\Delta t + \Delta x^2)$$

provided T_{tt} and T_{xxxx} are bounded. Hence the scheme is *consistent*. Next, using $T_{tt} = \alpha(T_{xx})_t = \alpha(T_t)_{xx} = \alpha^2 T_{xxxx}$, show that the choice $\mu = \alpha \Delta t / \Delta x^2 = 1 / 6$ reduces this error to $O(\Delta x^4)$.

PROBLEM 15. Show that if $\mu \le 1/2$ in (35) then the *local discretization error* satisfies $\|\mathbf{de}^{n+1}\| \le \|\mathbf{de}^n\| + \Delta t \cdot (A \Delta t + B \Delta x^2)$, where $\|\mathbf{de}^n\| = \max_{1 \le j \le M} |\mathbf{de}_j^n|$, $A = \max |T_{tt}/2|$, $B = \max |\alpha T_{xxxx}/12|$. Deduce that $\|\mathbf{de}^n\| \le n \Delta t (A \Delta t + B \Delta x^2)$, and since $n \Delta t \le t_{max}$ conclude that $\|\mathbf{de}^n\| = O(\Delta t + \Delta x^2)$, thus establishing

convergence of the scheme directly.

PROBLEM 16. (a) Show that if the CFL condition holds for (36) , then the error at any $n > 0$ due to initial roundoff error ε_j^0 is bounded by that initial error.
(b) Roundoff error may be introduced at every point that a computation is performed. Thus even if the CFL condition is met for (36) error is introduced not only at the initial step $n = 0$ but at every time step. What is its cumulative effect? Can it grow exponentially?

PROBLEM 17. (a) Derive the stability condition (40a) for imposed temperature at $x = 0$. (b) Derive the stability condition (40b) for imposed flux at $x = 0$.
(c) Derive the stability condition (40c) for the convective boundary condition at $x = 0$, and show that (40a) is sufficient for it.

PROBLEM 18. Combine (39) and (40) to establish (41-42).

PROBLEM 19. Analyze carefully the effect of using the discretization (43) for the first interior node. In particular, what happens if errors originate both at the initial line and at the boundary $j = 1$? What if no error originates at the boundary line?

PROBLEM 20. Find the counterpart to the implicit equation (43) for convective and flux boundary conditions.

PROBLEM 21. (a) Implement the explicit scheme (33) in a computer code (see **§4.1.I** for helpful suggestions) for the simple case (35) of uniform mesh and constant properties. To debug and validate your code, take $h = 0$ (whence the boundary conditions are $T_x(0, t) = T_x(l, t) = 0)$ and $T_{init}(x) = 100 \cos(\pi x / l)$, in which case the exact solution is $T(x, t) = \exp(-\pi^2 \alpha t / l^2 \cdot 100 \cos(\pi x / x)$, $0 \le x \le l$, $t \ge 0$. For simplicity, choose $l = 1$, $\alpha = 0.1$, $M = 10$ and compare the numerical and exact solutions up to time $t_{max} = 1$.
(b) Examine convergence by making runs with $M = 10, 20, 40, 80, 160$ nodes (remember to adjust Δt so that $\mu = 1/2$) and looking at the maximum error $\max_{0 \le t_n \le t_{max}} (\max_{1 \le j \le M} |T_j^n - T(x_j, t_n)|)$. Does it behave like $O(\Delta x^2)$? For which M do you get the least error? For that M, make a run in double precision. Does the error reduce further?
(c) Examine the effects of instability by fixing $M = 20$ and making runs with $\mu = \alpha \Delta t / \Delta x^2 = 0.4, 0.5, 0.501, 0.6, 1.0$. Discuss what you observe.
(d) According to PROBLEM 14, the choice $\mu = 1/6$ improves the error to $O(\Delta x^4)$. Test this by repeating (6) but with $\mu = 1/6$ now. Compare with the results from (6).

PROBLEM 22. For axially symmetric heat conduction in a hollow cylinder of unit height (PROBLEMS 3 and 9) with convective boundary condition at $r = R_{in}$ and imposed temperature at $r = R_{out}$: (a) set up the general $(0 \le \theta \le 1)$ algorithm (analogous to (31)); (b) find the stability conditions (analogous to (39, 40, 48)); (c) in the implicit case $(0 < \theta \le 1)$, write down the tridiagonal system (analogous to (47)).

PROBLEM 23. Do the same for spherically symmetric heat conduction in a sphere (PROBLEMS 4 and 10) with imposed temperature at $r = R_{out}$. In particular, examine carefully the discretization and stability restriction at the most internal node $[0, r_{1+1/2}]$. Note that the natural boundary condition at $r = 0$ is $q_{1/2} = 0$.

PROBLEM 24. Repeat PROBLEM 23 for the cases of imposed flux and of a convective boundary condition at $r = R_{out}$.

PROBLEM 25. Derive the discrete heat balance (see **§4.1.G**), for 2-dimensional (r, z), axially symmetric heat conduction in a cylinder, using the mesh of PROBLEM 5.

PROBLEM 26. Derive the discrete heat balance for 2-dimensional (r, θ), axially symmetric heat conduction in a sphere using the mesh of PROBLEM 6.

PROBLEM 27. Set up a 3-dimensional (r, θ, z) mesh for heat conduction in a hollow cylinder $R_{in} \le r \le R_{out}$, $0 \le \theta < 2\pi$, $0 \le z \le Z$, with $M_r \times M_\theta \times M_z$ nodes, and derive the discrete heat balance.

PROBLEM 28. For the process of PROBLEM 25, with convective boundary condition at $r = R_{in}$, imposed flux at $r = R_{out}$, and insulated axial faces $(z = 0, z = Z)$: (a) set up the computational algorithm for $0 \le \theta \le 1$; (b) find the stability conditions at internal and boundary nodes.

PROBLEM 29. Repeat, for the process of PROBLEM 26.

PROBLEM 30. Consider the initial-boundary value problem for the heat equation, $T_t = T_{xx}$, $a < x < b$, $t > 0$, with $T(x, 0) = C\cos(\gamma_0 x)$, $a < x < b$, and $T(a, t) = A\sin(\alpha_0 t)$, $T(b, t) = B\sin(\beta_0 t)$, where γ_0, α_0, β_0 are real numbers. Discuss how you would decide upon the size of the spatial and temporal mesh sizes to be used in calculating the solution to this problem.

PROBLEM 31. What would be your method for simulating heat transfer in a material whose thermal diffusivity varies by an order of magnitude or more over the range of temperatures encountered?

PROBLEM 32. (a) Discretize the heat equation $T_t = \alpha T_{xx}$ using the *centered*-difference $(T_j^{n+1} - T_j^{n-1}) / 2\Delta t$ for T_t and the standard centered-difference $(T_{j-1}^n - 2T_j^n + T_{j+1}^n) / \Delta x^2$ for T_{xx}. Show that this scheme is **unstable** for any $\mu > 0$!!! (b) In the previous scheme replace the central-term $-2T_j^n$ by $-(T_j^{n-1} + T_j^{n+1})$ to obtain the *Dufort-Frankel method* ([LAPIDUS-PINDER], [DUCHATEAU-ZACHMANN]). Show that if the ratio $\mu_1 := \Delta t / \Delta x$ is held fixed as $\Delta x, \Delta t \to 0$ then the method is consistent with the *hyperbolic* PDE $T_t + \alpha\mu_1 T_{tt} = \alpha T_{xx}$ and *not* with the heat equation.